

Chapter 13

Promoter Analysis: Gene Regulatory Motif Identification with A-GLAM

Leonardo Mariño-Ramírez, Kannan Tharakaraman, John L. Spouge, and David Landsman

Abstract

Reliable detection of *cis*-regulatory elements in promoter regions is a difficult and unsolved problem in computational biology. The intricacy of transcriptional regulation in higher eukaryotes, primarily in metazoans, could be a major driving force of organismal complexity. Eukaryotic genome annotations have improved greatly due to large-scale characterization of full-length cDNAs, transcriptional start sites (TSSs), and comparative genomics. Regulatory elements are identified in promoter regions using a variety of enumerative or alignment-based methods. Here we present a survey of recent computational methods for eukaryotic promoter analysis and describe the use of an alignment-based method implemented in the A-GLAM program.

Key Words: Promoter regions, transcription factor binding sites, enumerative methods, promoter comparison.

1. Introduction

The establishment and maintenance of temporal and spatial patterns of gene expression are achieved primarily by transcription regulation. Additionally, the precise control of timing and location of gene expression depends on the interaction between transcription factors and *cis*-acting sequence elements in promoter regions. Transcription factors can induce or repress gene expression upon binding of their cognate sequence element in the DNA. The discovery and categorization of the entire collection of transcription factor-binding sites (TFBSs) of an organism are among the greatest challenges in computational biology (1). Large-scale efforts involving genome mapping and identification of TFBS in

lower eukaryotes, such as the yeast *Saccharomyces cerevisiae*, have been successful (2). In contrast, similar efforts in vertebrates have proven difficult due to the presence of repetitive elements and an increased regulatory complexity (3–5).

The accurate prediction and identification of regulatory elements in higher eukaryotes remains a challenge for computational biology, despite recent progress in the development or improvement of different algorithms (6–19). Different strategies for motif recognition have been benchmarked to compare their performance (20). Typically, computational methods for identifying *cis*-regulatory elements in promoter sequences fall into two classes, enumerative and alignment techniques (21). We have developed algorithms that use enumerative approaches to identify *cis*-regulatory elements statistically significant over-represented in promoter regions (22). Subsequently, we developed an algorithm that combines both enumeration and alignment techniques to identify statistically significant *cis*-regulatory elements positionally clustered relative to a specific genomic landmark (23,24).

Promoter identification is the first step in the computational analysis that leads to the prediction of regulatory elements. In lower Eukaryotes this is a rather simple problem due to a relative high gene density with respect to the genome size. The yeast *Saccharomyces cerevisiae* has ~70% of its genome coding for proteins and its intergenic regions are fairly short (~440 bp in length) (25). In contrast, the human genome has a relative low gene density, with ~3% of the genome coding for proteins (26); this poses significant challenges for the identification of both the promoter and its regulatory elements. Despite the complexity of gene expression regulation in higher Eukaryotes (27), we now have a number of experimental and computational resources that can assist in the delineation of mammalian promoter regions. The experimental resources include full-length cDNA collections (28) and transcriptional start sites (TSS) (29). Additionally, complementary computational resources include the database of transcriptional start sites (DBTSS) (30) and promoter identification services (31–33). Many regulatory elements are located in the proximal promoter region (PPR) located a few hundred bases upstream the TSS (22) and the PPR can be generally defined by its low transposable element content (34).

The computational methods for the prediction and identification of transcription factor binding sites can be divided in two broad categories: algorithms for de novo identification and algorithms that recognize elements using prior knowledge of the elements. Enumerative and alignment methods form part of the de novo algorithms. Enumerative algorithms use exhaustive methods to examine exact DNA words of a fixed length to rank them according to a specific function that determine over-representation relative to a background distribution. An enumerative

method that estimates *p*-values with the standard normal approximation associated with *z*-scores (22) has been successfully applied for the identification of regulatory elements in higher Eukaryotes (35). Other enumerative methods include Weeder (16, 17), oligonucleotide frequency analysis (36), and QuickScore (14).

Alignment methods aim to identify functional elements by a multiple local alignment of all sequences of interest. The most popular algorithms in this category use an optimization procedure based in probabilistic sequence models to find statistically significant motifs; these include Gibbs sampling (37) or expectation maximization (11). Approaches that use a combination of enumerative and alignment methods have shown a significant improvement in the identification of regulatory elements in promoter sequences (23, 24).

Algorithms that use prior knowledge of known motifs often use position frequency matrices (PFMs) that contain the number of observed nucleotides at each position (38). Methods that assess statistical over-representation of known motifs in a set of sequences have been particularly successful (9). Additionally, motif scores determined by over-representation can be used as a proxy to perform promoter comparisons (39).

2. Program Usage

2.1. The A-GLAM Algorithm

The A-GLAM software package uses a Gibbs sampling algorithm to identify functional motifs in a set of sequences. Gibbs sampling, also known as successive substitution sampling, is a well-known Markov-chain Monte Carlo procedure for discovering sequence motifs (37). In brief, A-GLAM takes a set of sequences in FASTA format as input. The Gibbs sampling step in A-GLAM uses simulated annealing to maximize an “overall score,” corresponding to a Bayesian marginal log-odds score. The overall score is given by

$$s = \sum_{i=1}^w \left(\log_2 \frac{(\alpha - 1)!}{(c + \alpha - 1)!} + \sum_{(j)} \left\{ \log_2 \left[\frac{(c_{ij} + \alpha_j - 1)!}{(\alpha_j - 1)!} \right] - c_{ij} \log_2 p_j \right\} \right) \quad (1)$$

In equation (1), $m! = m(m - 1)\dots 1$ denotes a factorial; α_j , the pseudo-counts for nucleic acid *j* in each position; $\alpha = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4$, the total pseudo-counts in each position; c_{ij} , the count of nucleic acid *j* in position *i*; and $c = c_{i1} + c_{i2} + c_{i3} + c_{i4}$, the total number of aligned windows, which is independent of the position *i*. The underlying principle behind the overall score *s* in A-GLAM is explained in detail elsewhere (23).

The annealing maximization is initialized when A-GLAM places a single window of arbitrary size and position at every sequence, generating a gapless multiple alignment of the windowed subsequences. The program then proceeds through a series of iterations; on each iteration step, A-GLAM proposes a set of adjustments to the alignment. The proposal step is either a repositioning step or a resizing step. In a repositioning step, a single sequence is chosen uniformly at random from the alignment; and the set of adjustments include all possible positions in the sequence where the alignment window would fit without overhanging the ends of the sequence. In a resizing step, either the right or the left end of the alignment window is selected; and the set of proposed adjustments includes expanding or contracting the corresponding end of all alignment windows by one position at a time. Each adjustment leads to a different value of the overall score s . Then, A-GLAM accepts one of the adjustments randomly, with probability proportional to $\exp(s/T)$. A-GLAM may even exclude a sequence if doing so would improve alignment quality. The temperature T is gradually lowered to $T = 0$, with the intent of finding the gapless multiple alignments of the windows maximizing s . The maximization implicitly determines the final window size. The randomness in the algorithm helps it avoid local maxima and find the global maximum of s . However, due to the stochastic nature of the procedure, finding the optimum alignment it is not guaranteed.

In the default mode, A-GLAM repeats the annealing maximization procedure ten times from different starting points (ten runs). The rationale behind this is that if several of the runs converge to the same best alignment, the user has increased confidence that it is indeed the optimum alignment.

The individual score and its E-value in A-GLAM: The Gibbs sampling step produces an alignment whose overall score s is given by equation (1). Consider a window of length w that is about to be added to A-GLAM's alignment. Let $\delta_i(j)$ equal 1 if the window has nucleic acid j in position i , and 0 otherwise. The addition of the new window changes the overall score by

$$\Delta s = \sum_{i=1}^w \sum_{(j)} \delta_i(j) \left\{ \log_2 \left[\left(\frac{c_{ij} + \alpha_j}{c + \alpha} \right) / p_j \right] \right\} \quad (2)$$

The score change corresponds to scoring the new window according to a position specific scoring matrix (PSSM) that assigns the “individual score”

$$s_i(j) = \log_2 \left[\left(\frac{c_{ij} + \alpha_j}{c + \alpha} \right) / p_j \right] \quad (3)$$

to nucleic acid j in position i . Equation (3) represents a log-odds score for nucleic acid j in position i under an alternative hypothesis with probability $(c_{ij} + \alpha_j)/(c + \alpha)$ and a null hypothesis with

probability p_{ij} . PSI-BLAST (40) uses equation (3) to calculate E-values. The derivation through equation (2) confirms the PSSM in equation (3) as the natural choice for evaluating individual sequences.

The assignment of an E-value to a subsequence with a particular individual score is done as follows. Consider the alignment sequence containing the subsequence. Let n be the sequence length, and recall that w is the window size. If ΔS_i denotes the quantity in equation (2) if the final letter in the window falls at position i of the alignment sequence, then $\Delta S^* = \max\{\Delta S_i : i = w, \dots, n\}$ is the maximum individual score over all sequence positions i . We assigned an E-value to the actual value $\Delta S^* = \Delta s^*$, as follows. Staden's method (41) yields $\mathbb{P}\{\Delta S_i \geq \Delta s^*\}$ (independent of i) under the null hypothesis of bases chosen independently and randomly from the frequency distribution $\{p_j\}$. The E-value $E = (n - w + 1)\mathbb{P}\{\Delta S_i \geq \Delta s^*\}$ is therefore the expected number of sequence positions with an individual score exceeding Δs^* . The factor $n - w + 1$ in E is essentially a multiple test correction.

More recently, the A-GLAM package has been improved to allow the identification of multiple instances of an element within a target sequence (24). The optional “scanning step” after Gibbs sampling produces a PSSM given by equation (3). The new scanning step resembles an iterative PSI-BLAST search based on the PSSM (**Fig. 13.1**). First, it assigns an “individual score” to each subsequence of appropriate length within the input sequences using the initial PSSM. Second, it computes an E-value from each individual score to assess the agreement between the corresponding subsequence and the PSSM. Third, it permits subsequences with E-values falling below a threshold to contribute to the underlying PSSM, which is then updated using the Bayesian calculus. A-GLAM iterates its scanning step to convergence, at which point no new subsequences contribute to the PSSM. After convergence, A-GLAM reports predicted regulatory elements within each sequence in order of increasing E-values; users then have a statistical evaluation of the predicted elements in a convenient presentation. Thus, although the Gibbs sampling step in A-GLAM finds at most one regulatory element per input sequence, the scanning step can now rapidly locate further instances of the element in each sequence.

2.2. Hardware

The minimum hardware requirements are a personal computer with at least 512 MB of random access memory (RAM) connected to the Internet as well as access to a Linux or UNIX workstation where A-GLAM will be installed. The connectivity between the personal computer and the workstation is typically established by the Secure Shell (SSH) protocol, a widely used open source of the protocol available at <http://www.openssh.org/>.

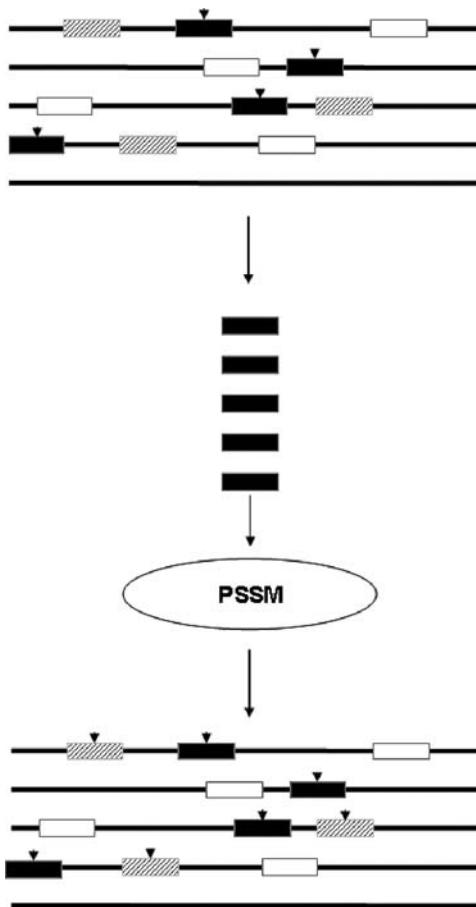


Fig. 13.1. Strategy to find multiple motif instances in A-GLAM. The Gibbs sampling identifies up to one motif per sequence (indicated by a *black box* and an *arrowhead*). The sequences are then used to construct a position specific score matrix (PSSM) that is used iteratively to discover multiple motif instances per sequence (indicated by *dashed boxes*).

2.3. Software

A modern version of the Perl programming language installed on the Linux or UNIX workstation freely available at <http://www.perl.com/> will allow the user to parse A-GLAM's output. The A-GLAM package (23) freely available at <http://ftp.ncbi.nih.gov/pub/spouge/papers/archive/AGLAM/> is currently available as source code and binary packages for the Linux operating system.

Installation of the Linux binary: get the executable from the FTP site and set execute permissions.

```
$ chmod +x aglam
```

Installation from source: unpack the glam archive in a convenient location and compile A-GLAM.

```
$tar -zxvf aglam.tar.gz
$cd aglam
$make aglam
```

Then you could place the binary in your path: \$HOME/bin or /usr/local/bin/.

2.4. Data Files

A-GLAM accepts input data in FASTA format containing the sequences to be analyzed. The FASTA format consists of one or more sequences identified by a line beginning with the “>” character that should include a unique identifier and a short description about the sequence. The next line(s) should contain the sequence string. A-GLAM expects the standard nucleic acid IUPAC code.

2.5. A-GLAM Options

Some important options to modify A-GLAM’s behavior are described below:

```
$aglam <fasta_file.fa>
```

This command simply uses the standard Gibbs sampling procedure to find sequence motifs in “fasta_file.fa.”

```
$aglam <fasta_file.fa> -n 30000 -a 8 -b 16 -j
```

These sets of commands instruct the program to search only the given strand of the sequences to find motifs of length between 8 and 16 bp. The flag *n* specifies the number of iterations performed in each of the ten runs. Low values of *n* are adequate when the problem size is small, i.e., when the sequences are short and more importantly there are few of them, but high values of *n* are needed for large problems. In addition, smaller values of *n* are sufficient when there is a strong alignment to be found, but larger values are necessary when there is no strong alignment, e.g., for finding the optimal alignment of random sequences. You will have to choose *n* on a case-by-case basis. This parameter also controls the tradeoff between speed and accuracy.

```
$aglam <fasta_file.fa> -i TATA
```

This important option sets the program to run in a “seed” oriented mode. The above command restricts the search to sequences that are TATA-like. Unlike the procedure followed in the standard Gibbs sampling algorithm, however, A-GLAM continues to align one exact copy of the “seed” in all “seed sequences.” Therefore, A-GLAM uses the seed sequences to direct its search in the remaining non-seed “target sequences.” Using this option leads to the global optimum quickly.

```
$aglam <fasta_file.fa> -l -k 0.05 -g 2000
```

Usable only with version 1.1. This set of commands instructs the program to find multiple motif instances in each input sequence via the scanning step (described above). Those instances that receive an E-value less than 0.05 are included in the PSSM. The search for multiple motifs is carried on until either (a) no new motifs are present or (b) the user-specified number of iterations (in this case, it is 2000) is attained, whichever comes first.

3. Example

The A-GLAM package includes documentation and test datasets. Here, we will use a dataset obtained from a large-scale chromatin immunoprecipitation in *Saccharomyces cerevisiae* (2), combined with DNA microarrays (42) to detect interactions between transcription factors and a DNA sequence *in vivo*. The DNA sequence binding specificity of various transcription factors can then be inferred using A-GLAM on intergenic regions bound by a particular transcription factor. Here, we will use the intergenic regions bound by Snt2p (*see Note 1*).

3.1. Promoter Identification

The *Saccharomyces* Genome Database (SGD) maintains the most current annotations of the yeast genome (*see* <http://www.yeast-genome.org/>). The SGD FTP site contains the DNA sequences annotated as intergenic regions in FASTA format (available at ftp://genome-ftp.stanford.edu/pub/yeast/sequence/genomic_sequence/intergenic/), indicating the 5' and 3' flanking features. Additionally, a tab delimited file with the annotated features of the genome is necessary to determine the orientation of the intergenic regions relative to the genes (available at ftp://genome-ftp.stanford.edu/pub/yeast/chromosomal_feature/). The two files can be used to extract upstream intergenic regions. Additionally, there are a number of Web services that facilitate the identification of proximal promoter in mammalian genomes; these include TRED (32), EPD (33), and Promoser (31).

3.2. Identification of cis-Regulatory Elements in Promoter Regions

Construct FASTA files for each of the promoters to be included in the analysis. The Perl programming language can be used in conjunction with BioPerl libraries (freely available at <http://www.bioperl.org/>) to generate files in FASTA format. In this particular example all relevant files can be found on the Fraenkel Web site at http://fraenkel.mit.edu//Harbison/release_v24.

The A-GLAM package has a number of options that can be used to adjust search parameters.

```
$aglam
Usage summary: aglam [ options] myseqs.fa
Options:
-h help: print documentation
-n end each run after this many iterations
without improvement (10000)
-r number of alignment runs (10)
-a minimum alignment width (3)
-b maximum alignment width (10000)
-j examine only one strand
-i word seed query ()
-f input file containing positions of the
motifs ()
-z turn off ZOOPS (force every sequence to
participate in the alignment)
-v print all alignments in full
-e turn off sorting individual sequences in an
alignment on p-value
-q pretend residue abundances = 1/4
-d frequency of width-adjusting moves (1)
-p pseudocount weight (1.5)
-u use uniform pseudocounts: each pseudocount =
p/4
-t initial temperature (0.9)
-c cooling factor (1)
-m use modified Lam schedule (default = geo-
metric schedule)
-s seed for random number generator (1)
-w print progress information after each
iteration
-l find multiple instances of motifs in each
sequence
-k add instances of motifs that satisfy the
cutoff e-value (0)
-g number of iterations to be carried out in
the post-processing step (1000)
```

Run A-GLAM to identify regulatory elements present in the promoter regions bound by Snt2p. A-GLAM uses sequences in FASTA format as input. There are 46 intergenic regions bound by Snt2p that were identified by ChIP-chip in a large-scale study (2). These regions vary in length from 71 to 1,512 bp with an average of 398 bp. A-GLAM is able to identify statistically significant motifs for Snt2p and rank them according to their

individual *p*-values. A-GLAM has a number of useful command line options that can be adjusted to improve ab initio motif finding; in this example we have restricted the search to motifs no larger than 20 bp and instructed the program to find multiple instances of motifs in each sequence using a strategy that resembles an iterative PSI-BLAST search based on the PSSM constructed by the Gibbs sampling step (24). The output of the A-GLAM program is presented in Fig. 13.2. In the default mode, A-GLAM repeats the annealing maximization procedure ten times from different starting points (ten runs). The rationale behind this is that if several of the runs converge to the same best alignment, the user has increased confidence that it is indeed the optimum alignment. The user can adjust the number of alignment runs by setting the *-r* flag (see Note 2). The number of iterations can also be adjusted for large datasets. The default value is set at 10,000 without alignment improvement, using the *-n* flag the number of iterations can be increased to extend coverage of the sequence space.

A-GLAM identifies candidate sequences that could serve as Snt2p binding sites. The candidate sequences found by A-GLAM are in agreement with previous findings where other motif finding algorithms were used (2) and Fig. 13.3. Additional examples where we have successfully used A-GLAM to complement experimental efforts for the identification of regulatory elements include motifs for Spt10p in yeast and the CREB-binding protein (34, 35). In this particular example, the program constructs a PSSM using the sequences from the optimal alignment to find multiple instances (see Note 3). The multiple alignments produced by A-GLAM can be represented graphically by sequence logos (43, 44) (see Note 4).

4. Notes



1. The primary data can be obtained from the Fraenkel Laboratory Web site at <http://fraenkel.mit.edu/Harbison/>.
2. The number of alignment runs is 10 by default; however, the user can increase the number of runs to boost the confidence of the results. The user has the option *-v* to print all alignments generated in each run; by default A-GLAM will report only the highest scoring alignment.
3. Alternatively, the user could run A-GLAM without the *-l* flag and construct a position frequency matrix that in turn could be used to scan the target sequences for additional instances of the motif. The TFBS Perl modules for

```

$ aglam -b 20 -l SNT2_YPD.fsa
A-GLAM: Anchored Gapless Local Alignment of Multiple Sequences
Compiled on Feb 9 2008
aglam -l SNT2_YPD.fsa

Run 1... 25340 iterations
Run 2... 26770 iterations
Run 3... 22597 iterations
Run 4... 17786 iterations
Run 5... 23816 iterations
Run 6... 42556 iterations
Run 7... 19556 iterations
Run 8... 22526 iterations
Run 9... 23310 iterations
Run 10... 21531 iterations

! The sequence file was [SNT2_YPD.fsa]
! Reading the file took [0] secs
! Sequences in file [46]
! Maximum possible alignment width [142]
! Score [400] bits
! Motif Width [12]
! Runs [10]

! Best possible alignment:

>iYNL182C 6.2046e-10          202 ATGGCGCTATCA 213 + (10.24060) (1.714353e-02)
>iYBL075C 7.9181e-10          278 CGGGCGCTATCA 267 - (12.62630) (3.136227e-04)
>iYILL160C 1.0190e-09          211 ACGGCGCTACCA 222 + (14.16730) (2.230312e-05)
                                         208 AAGGCCTATCA 197 - (10.97000) (4.259169e-03)
>iYPR183W 1.6110e-09          237 CGGGCGCTACCA 248 + (14.39810) (8.284745e-06)
>iYCR090C-1 2.2463e-09          575 ACGGCGCTATCA 564 - (12.39550) (1.190739e-03)
>iYAL039C-0 5.6844e-09          281 CGGGCGCTACCA 292 + (14.39810) (2.092311e-05)
>iYPR157W 1.0834e-08          343 GTGGCGCTATCA 332 - (10.47150) (1.119393e-02)
                                         absent
>iYLR149C 1.4205e-08          252 ATGGCGCTACCA 263 + (12.01250) (1.704126e-03)
>iYJL093C 3.7648e-08          279 CGGGCGCTATCA 268 - (12.62630) (6.283269e-04)
>iYBR143C 2.6501e-07          202 ACGGCGCTATCA 213 + (12.39550) (1.581019e-03)
>iYLR176C 1.6035e-06          221 GTGGCGCTACCA 232 + (12.24330) (1.517043e-03)
>iYPR104C 6.0302e-06          420 ATGGCGCTATCA 431 + (10.24060) (1.333119e-02)
>iYBR138C 9.2799e-06          203 CGGGCGCTAGCA 214 + (12.42200) (4.809407e-04)
                                         206 CCGGCTCGGCCA 195 - (8.225040) (4.975689e-02)
>tP(UGG)M 1.4586e-05          26 CCAGCTCGCCCC 15 - (8.644490) (1.269803e-02)
                                         99 ACCACTAGACCA 110 + (7.042530) (4.773258e-02)
                                         absent
>iYHR217C 1.7438e-05          145 TCGGCCTACCA 134 - (11.23880) (3.713288e-03)
>iYHR138C 3.9997e-05          absent
                                         absent
>iYKL172W 4.5759e-05          absent
>IntYGL103W 4.7991e-05          21 ACCACTCGGCCA 10 - (9.819350) (3.647425e-03)
>tL(UAG)L2 4.9893e-05          absent
>iYJR152W 5.1753e-05          35 CCTGGCGGGCA 46 + (9.031130) (6.321172e-03)
>tS(AGA)M 6.7229e-05          81 AAAGCTCTACCA 92 + (10.76890) (1.315551e-03)
                                         absent
>tI(UAU)L 9.5693e-05          26 GCAACGCGACCG 15 - (8.373710) (1.822708e-02)
>iYCR090C-0 1.0515e-04          absent
                                         absent
>IntYPL081W 1.1828e-04          162 CCGATTGACCA 151 - (7.925580) (4.569485e-02)
>SNR190 1.1910e-04          24 ACCGCTCGGCCA 13 - (11.62350) (5.732354e-04)
>tP(AGG)C 1.3420e-04          24 CCGGCTCGCCCC 13 - (9.501020) (4.510700e-03)
                                         absent
>tS(GCU)U 1.9681e-04          absent
                                         absent
>tH(GUG)M 2.0224e-04          absent
>SNR43 2.6811e-04          71 CCAGCGCGGGCA 60 - (10.69280) (1.375755e-03)
                                         absent
>tK(UUU)P 3.1249e-04          55 AACGCTCTACCA 66 + (10.63040) (1.330965e-03)
>tN(GUU)P 4.7144e-04          32 CCAAATTGGCCA 21 - (7.907740) (2.015338e-02)
>tV(AAC)M3 4.8949e-04          60 CCGACTAGACCA 71 + (7.828140) (1.674674e-02)
>tA(UGC)O 5.7662e-04          48 AGCCGCTATCA 59 + (9.775590) (2.504172e-03)
>tT(AGU)O2 6.8638e-04          60 CCAAATTGGCCA 71 + (7.907740) (1.737360e-02)
>tR(UCU)M2 7.2321e-04          55 GACGGCTTGGCCA 66 + (9.845220) (2.902318e-03)
                                         absent
>iYLR228C-1 8.1088e-04          absent
>tQ(UUG)L 8.1134e-04          absent
>tC(GCA)P2 9.1920e-04          46 GCTGGCTACCA 57 + (11.88000) (2.284798e-04)
>iYDR261C-1 9.4038e-04          absent
                                         absent
>SNR44 9.4060e-04          26 CCAACGTTGCCA 37 + (9.164880) (5.191352e-03)

! 34 sequences in alignment
! Residue abundances:Pseudocounts
! A = 0.311204:0.466806 C = 0.188796:0.283194 G = 0.188796:0.283194 T = 0.311204:0.466806
! Total Time to find best alignment [13.92] secs

```

Fig. 13.2. A-GLAM output for a set of sequences containing an SNT2p motif identified using ChIP-chip. A-GLAM works by analyzing completely random alignment of the sequences and making small refinements over ten alignment runs with many iterations.

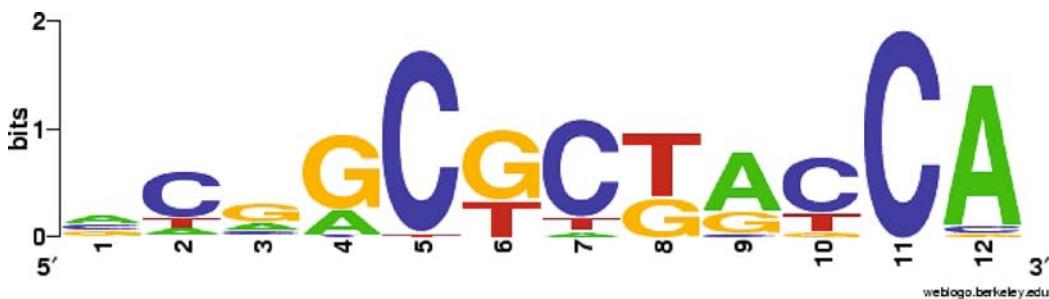


Fig. 13.3. Snt2p regulatory motif identified with A-GLAM. Sequence logo representation of the motif obtained from the ungapped multiple sequence alignment identified by A-GLAM.

transcription factor binding detection and analysis provide a flexible and powerful framework (available at <http://tfbs.genereg.net/>).

4. Other Web servers for logo generation include enoLOGOS (available on the Web at <http://biodev.hgen.pitt.edu/enologos/>) and Pictogram (<http://genes.mit.edu/pictogram.html>).

Acknowledgments

This research was supported by the Intramural Research Program of the NIH, NLM, NCBI.

References

1. Elnitski, L., Jin, V. X., Farnham, P. J., and Jones, S. J. (2006) Locating mammalian transcription factor binding sites: a survey of computational and experimental techniques. *Genome Res* **16**, 1455–64.
2. Harbison, C. T., Gordon, D. B., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., Danford, T. W., Hannett, N. M., Tagne, J. B., Reynolds, D. B., Yoo, J., et al. (2004) Transcriptional regulatory code of a eukaryotic genome. *Nature* **431**, 99–104.
3. Bieda, M., Xu, X., Singer, M. A., Green, R., and Farnham, P. J. (2006) Unbiased location analysis of E2F1-binding sites suggests a widespread role for E2F1 in the human genome. *Genome Res* **16**, 595–605.
4. Cawley, S., Bekiranov, S., Ng, H. H., Kapranov, P., Sekinger, E. A., Kampa, D., Piccolboni, A., Sementchenko, V., Cheng, J., Williams, A. J., et al. (2004) Unbiased mapping of transcription factor binding sites along human chromosomes 21 and 22 points to widespread regulation of noncoding RNAs. *Cell* **116**, 499–509.
5. Guccione, E., Martinato, F., Finocchiaro, G., Luzi, L., Tizzoni, L., Dall’Olio, V., Zardo, G., Nervi, C., Bernard, L., and Amati, B. (2006) Myc-binding-site recognition in the human genome is determined by chromatin context. *Nat Cell Biol* **8**, 764–70.
6. Hughes, J. D., Estep, P. W., Tavazoie, S., and Church, G. M. (2000) Computational identification of *cis*-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J Mol Biol* **296**, 1205–14.

7. Workman, C. T., and Stormo, G. D. (2000) ANN-Spec: a method for discovering transcription factor binding sites with improved specificity. *Pac Symp Biocomput* **5**, 467–78.
8. Hertz, G. Z., and Stormo, G. D. (1999) Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics* **15**, 563–77.
9. Frith, M. C., Fu, Y., Yu, L., Chen, J. F., Hansen, U., and Weng, Z. (2004) Detection of functional DNA motifs via statistical over-representation. *Nucleic Acids Res* **32**, 1372–81.
10. Ao, W., Gaudet, J., Kent, W. J., Muttumu, S., and Mango, S. E. (2004) Environmentally induced foregut remodeling by PHA-4/FoxA and DAF-12/NHR. *Science* **305**, 1743–6.
11. Bailey, T. L., and Elkan, C. (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int Conf Intell Syst Mol Biol* **2**, 28–36.
12. Eskin, E., and Pevzner, P. A. (2002) Finding composite regulatory patterns in DNA sequences. *Bioinformatics* **18 Suppl 1**, S354–63.
13. Thijss, G., Lescot, M., Marchal, K., Rombauts, S., De Moor, B., Rouze, P., and Moreau, Y. (2001) A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics* **17**, 1113–22.
14. Régnier, M., and Denise, A. (2004) Rare events and conditional events on random strings. *Discrete Math Theor Comput Sci* **6**, 191–214.
15. Favorov, A. V., Gelfand, M. S., Gerasimova, A. V., Raycheev, D. A., Mironov, A. A., and Makeev, V. J. (2005) A Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length. *Bioinformatics* **21**, 2240–5.
16. Pavese, G., Mereghetti, P., Zambelli, F., Stefani, M., Mauri, G., and Pesole, G. (2006) MoD Tools: regulatory motif discovery in nucleotide sequences from co-regulated or homologous genes. *Nucleic Acids Res* **34**, W566–70.
17. Pavese, G., Zambelli, F., and Pesole, G. (2007) WeederH: an algorithm for finding conserved regulatory motifs and regions in homologous sequences. *BMC Bioinformatics* **8**, 46.
18. Sinha, S., and Tompa, M. (2003) YMF: A program for discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Res* **31**, 3586–8.
19. Blanchette, M., Bataille, A. R., Chen, X., Poitras, C., Laganier, J., Lefebvre, C., Deblois, G., Giguere, V., Ferretti, V., Bergeron, D., et al. (2006) Genome-wide computational prediction of transcriptional regulatory modules reveals new insights into human gene expression. *Genome Res* **16**, 656–68.
20. Tompa, M., Li, N., Bailey, T. L., Church, G. M., De Moor, B., Eskin, E., Favorov, A. V., Frith, M. C., Fu, Y., Kent, W. J., et al. (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotechnol* **23**, 137–44.
21. Ohler, U., and Niemann, H. (2001) Identification and analysis of eukaryotic promoters: recent computational approaches. *Trends Genet* **17**, 56–60.
22. Marino-Ramirez, L., Spouge, J. L., Kang, G. C., and Landsman, D. (2004) Statistical analysis of over-represented words in human promoter sequences. *Nucleic Acids Res* **32**, 949–58.
23. Tharakaraman, K., Marino-Ramirez, L., Sheetlin, S., Landsman, D., and Spouge, J. L. (2005) Alignments anchored on genomic landmarks can aid in the identification of regulatory elements. *Bioinformatics* **21 Suppl 1**, i440–8.
24. Tharakaraman, K., Marino-Ramirez, L., Sheetlin, S., Landsman, D., and Spouge, J. L. (2006) Scanning sequences after Gibbs sampling to find multiple occurrences of functional elements. *BMC Bioinformatics* **7**, 408.
25. Goffeau, A., Barrell, B. G., Bussey, H., Davis, R. W., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J. D., Jacq, C., Johnston, M., et al. (1996) Life with 6000 genes. *Science* **274**, 546, 563–47.
26. Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., et al. (2001) Initial sequencing and analysis of the human genome. *Nature* **409**, 860–921.
27. Levine, M., and Tjian, R. (2003) Transcription regulation and animal diversity. *Nature* **424**, 147–51.
28. Carninci, P., Waki, K., Shiraki, T., Konno, H., Shibata, K., Itoh, M., Aizawa, K., Arakawa, T., Ishii, Y., Sasaki, D., et al. (2003) Targeting a complex transcriptome: the construction of the mouse full-length cDNA encyclopedia. *Genome Res* **13**, 1273–89.
29. Kimura, K., Wakamatsu, A., Suzuki, Y., Ota, T., Nishikawa, T., Yamashita, R.,

- Yamamoto, J., Sekine, M., Tsuritani, K., Wakaguri, H., et al. (2006) Diversification of transcriptional modulation: large-scale identification and characterization of putative alternative promoters of human genes. *Genome Res* **16**, 55–65.
30. Suzuki, Y., Yamashita, R., Sugano, S., and Nakai, K. (2004) DBTSS, DataBase of Transcriptional Start Sites: progress report 2004. *Nucleic Acids Res* **32**, D78–81.
 31. Halees, A. S., and Weng, Z. (2004) Promoter: improvements to the algorithm, visualization and accessibility. *Nucleic Acids Res* **32**, W191–4.
 32. Jiang, C., Xuan, Z., Zhao, F., and Zhang, M. Q. (2007) TRED: a transcriptional regulatory element database, new entries and other development. *Nucleic Acids Res* **35**, D137–40.
 33. Schmid, C. D., Perier, R., Praz, V., and Bucher, P. (2006) EPD in its twentieth year: towards complete promoter coverage of selected model organisms. *Nucleic Acids Res* **34**, D82–5.
 34. Eriksson, P. R., Mendiratta, G., McLaughlin, N. B., Wolfsberg, T. G., Marino-Ramirez, L., Pompa, T. A., Jainerin, M., Landsman, D., Shen, C. H., and Clark, D. J. (2005) Global regulation by the yeast Spt10 protein is mediated through chromatin structure and the histone upstream activating sequence elements. *Mol Cell Biol* **25**, 9127–37.
 35. Riz, I., Akimov, S. S., Eaker, S. S., Baxter, K. K., Lee, H. J., Marino-Ramirez, L., Landsman, D., Hawley, T. S., and Hawley, R. G. (2007) TLX1/HOX11-induced hematopoietic differentiation blockade. *Oncogene* **26**, 4115–23.
 36. van Helden, J., Andre, B., and Collado-Vides, J. (1998) Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J Mol Biol* **281**, 827–42.
 37. Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F., and Wootton, J. C. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* **262**, 208–14.
 38. Wasserman, W. W., and Sandelin, A. (2004) Applied bioinformatics for the identification of regulatory elements. *Nat Rev Genet* **5**, 276–87.
 39. Marino-Ramirez, L., Jordan, I. K., and Landsman, D. (2006) Multiple independent evolutionary solutions to core histone gene regulation. *Genome Biol* **7**, R122.
 40. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* **25**, 3389–402.
 41. Staden, R. (1989) Methods for calculating the probabilities of finding patterns in sequences. *Comput Appl Biosci* **5**, 89–96.
 42. Ren, B., Robert, F., Wyrick, J. J., Aparicio, O., Jennings, E. G., Simon, I., Zeitlinger, J., Schreiber, J., Hannett, N., Kanin, E., et al. (2000) Genome-wide location and function of DNA binding proteins. *Science* **290**, 2306–9.
 43. Schneider, T. D., and Stephens, R. M. (1990) Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res* **18**, 6097–100.
 44. Crooks, G. E., Hon, G., Chandonia, J. M., and Brenner, S. E. (2004) WebLogo: a sequence logo generator. *Genome Res* **14**, 1188–90.